



TUGAS AKHIR - TF 141581

**OPTIMISASI PEMISAHAN H_2S PADA
REGENERATOR COLUMN DI ACID GAS
REMOVAL UNIT DENGAN METODE
GENETIC ALGORITHM**

HAFIZH GHAZIDIN
NRP 2413 100 100

Dosen Pembimbing
Hendra Cordova, S.T., M.T.

DEPARTEMEN TEKNIK FISIKA
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2017



FINAL PROJECT - TF 141581

***OPTIMIZATION OF H_2S SEPARATION ON
REGENERATOR COLUMN OF ACID GAS
REMOVAL UNIT BY USING GENETIC
ALGORITHM METHOD***

HAFIZH GHAZIDIN
NRP 2413 100 100

Supervisor
Hendra Cordova, S.T., M.T.

DEPARTMENT OF ENGINEERING PHYSICS
Faculty of Industrial Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2017

PERNYATAAN BEBAS PLAGIASI

Saya yang bertanda tangan di bawah ini:

Nama : Hafizh Ghazidin

NRP : 2413100100

Departemen : Teknik Fisika FTI-ITS

dengan ini menyatakan bahwa tugas akhir saya berjudul *OPTIMISASI PEMISAHAN H_2S PADA REGENERATOR COLUMN DI ACID GAS REMOVAL UNIT DENGAN METODE GENETIC ALGORITHM* adalah bebas dari plagiasi. Apabila pernyataan ini terbukti tidak benar, maka saya bersedia menerima sanksi sesuai ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, 20 Juli 2017

Yang membuat pernyataan,



Hafizh Ghazidin

LEMBAR PENGESAHAN

OPTIMISASI PEMISAHAN H_2S PADA REGENERATOR COLUMN DI ACID GAS REMOVAL UNIT DENGAN METODE GENETIC ALGORITHM

TUGAS AKHIR

Oleh :

Hafizh Ghazidin
NRP : 2413100100

Surabaya, Juli 2017
Mengetahui/Menyetujui

Pembimbing I



Hendra Cordova, S.T., M.T.
NIPN. 196905301994121001

Ketua Departemen
Teknik Fisika FTI-ITS



Agus Muhammad Hatta, S.T., M.Si., Ph.D.
NIPN. 197809022003121002

LEMBAR PENGESAHAN

OPTIMISASI PEMISAHAN H_2S PADA REGENERATOR COLUMN DI ACID GAS REMOVAL UNIT DENGAN METODE GENETIC ALGORITHM

TUGAS AKHIR


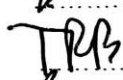
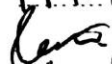

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Teknik
pada
Bidang Studi Rekayasa Instrumentasi dan Kontrol
Program Studi S-1 Departemen Teknik Fisika
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember

Oleh :

HAFIZH GHAZIDIN

NRP. 2413100100

Disetujui oleh Tim Penguji Tugas Akhir :

1. Hendra Cordova, S.T., M.T.  (Pembimbing I)
2. Totok Ruki Biyanto, Ph.D.  (Ketua Penguji)
3. Dr. Katherin Indriawati, S.T., M.T.  (Penguji I)
4. Lizda Johar Mawarani, S.T., M.T.  (Penguji II)

SURABAYA

JULI, 2017

OPTIMISASI PEMISAHAN H_2S PADA *REGENERATOR COLUMN* DI *ACID GAS REMOVAL UNIT* DENGAN METODE *GENETIC ALGORITHM*

Nama Mahasiswa : Hafizh Ghazidin
NRP : 2413 100 100
Departemen : Teknik Fisika FTI-ITS
Dosen Pembimbing : Hendra Cordova, S.T., M.T.

Abstrak

Dalam pengolahan gas alam terdapat proses menghilangkan kandungan H_2S dari *feed gas*. H_2S perlu dihilangkan karena merupakan zat asam yang berbahaya dan dapat menurunkan kualitas dari gas alam. Proses penghilangan H_2S dilakukan menggunakan DEAmine. DEAmine akan menyerap H_2S yang terdapat pada *feed gas* di dalam *absorber*. Setelah itu DEAmine akan diregenerasi pada *regenerator column*. Pada proses di *regenerator column*, H_2S akan dipisahkan dari DEAmine dengan bantuan *reboiler* untuk pemanasannya. Agar pemisahan H_2S dapat berjalan optimal maka perlu ditentukan laju aliran *hot oil reboiler* yang paling baik karena fraksi H_2S pada *top product* akan dipengaruhi oleh laju aliran *hot oil reboiler*. Pada penelitian ini, dilakukan optimisasi untuk mengoptimalkan pemisahan H_2S pada *regenerator column*. Proses di *regenerator column* dimodelkan menggunakan HYSYS. Kemudian dilakukan simulasi untuk menemukan *gain plant*. *Gain plant* tersebut akan digunakan untuk mendapatkan laju aliran *hot oil reboiler* yang paling baik melalui optimisasi dengan metode *genetic algorithm*. Hasil yang didapatkan dari penelitian ini adalah nilai laju aliran *hot oil reboiler* yang paling baik sebesar 15216823 kJ/hr dengan nilai temperatur dan fraksi H_2S masing-masing sebesar 106,22 °C dan 0,018508.

Kata Kunci: H_2S , *regenerator column*, *gain plant*, *genetic algorithm*

**OPTIMIZATION OF H_2S SEPARATION IN
REGENERATOR COLUMN OF ACID GAS REMOVAL
UNIT BY GENETIC ALGORITHM METHOD**

Name : **Hafizh Ghazidin**
NRP : **2413 100 100**
Department : **Engineering Physics FTI-ITS**
Supervisor : **Hendra Cordova, S.T., M.T.**

Abstract

In the processing of natural gas, there was a process for removing H_2S from the feed gas. H_2S needs to be removed because it is an acid gas that can reduce the quality of the product. The process of removing H_2S is done by using DEAmine. DEAmine absorbed H_2S that contained in the feed gas in the absorber. Then, DEAmine regenerated in the regenerator column. On the process in the regenerator column, H_2S is separated from DEAmine by using reboiler for heating. For optimal separation of H_2S , it is necessary to determine the best hot oil flow rate on the reboiler because the fraction of H_2S on to product is influenced by the hot oil flow rate on the reboiler. In this study, optimization is done to optimize the separation of H_2S in the regenerator column. The process in the regenerator column is modeled by using HYSYS. Then, the simulation is done to find the gain plant. The gain plant is used to find the best hot oil flow rate on the reboiler through optimization by genetic algorithm method. The results of this study are the best hot oil flow rate on the reboiler is 15216823 kJ/hr, with the value of the temperature and the fraction of H_2S are 106,22 °C and 0,018508.

Keywords: ***H_2S , regenerator column, gain plant, genetic algorithm***

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT, karena rahmat dan hikmat-Nya sehingga penulis diberikan kesehatan, kemudahan, dan kelancaran dalam menyusun laporan tugas akhir ini. Tidak lupa juga penulis menyampaikan ucapan terima kasih kepada keluarga dan para sahabat. Oleh karena dukungan mereka, penulis mampu menyusun laporan tugas akhir yang berjudul:

“OPTIMISASI PEMISAHAN H_2S PADA REGENERATOR COLUMN DI ACID GAS REMOVAL UNIT DENGAN METODE GENETIC ALGORITHM”

Tugas akhir ini merupakan salah satu persyaratan akademik yang harus dipenuhi dalam Program Studi S-1 Teknik Fisika FTI-ITS. Penulis menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Hendra Cordova, S.T., M.T. selaku dosen pembimbing tugas akhir ini, yang selalu memberikan bimbingan dan semangat pada penulis.
2. Totok Ruki Biyanto, Ph.D. selaku ketua bidang minat rekayasa instrumentasi dan kontrol, yang telah banyak membantu penulis dalam menyelesaikan tugas akhir ini.
3. Agus M. Hatta, S.T., M.Si., Ph.D. selaku ketua departemen Teknik Fisika ITS.
4. Ir. Matraji, M.Kom. selaku dosen wali penulis.
5. Robertus Raditya dan Achmad Reza selaku teman-teman satu dosen pembimbing yang telah banyak memotivasi penulis untuk menyelesaikan tugas akhir ini.
6. Januar Ananta dan Kevin Sanjoyo selaku teman-teman seperjuangan tugas akhir, yang telah banyak membantu penulis dalam menyelesaikan tugas akhir ini.
7. Lukas Rudy Paembong, Ilham Zarkasie, dan Irham Raditya selaku teman-teman dari departemen Teknik Kimia ITS, yang telah banyak membantu penulis dalam menyelesaikan tugas akhir ini.

Penulis menyadari bahwa mungkin masih ada kekurangan dalam laporan ini, sehingga kritik dan saran penulis terima. Semoga laporan ini dapat berguna dan bermanfaat bagi penulis dan pihak yang membacanya.

Surabaya, 20 Juli 2017

Penulis

DAFTAR ISI

ABSTRAK	i
ABSTRACT	iii
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL	xi
DAFTAR NOTASI	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	3
1.4 Lingkup Kerja	3
BAB II DASAR TEORI	5
2.1 Pemurnian Gas	5
2.2 <i>Regenerator Column</i>	6
2.3 Peng-Robinson <i>Equation of State</i>	11
2.4 Pendekatan FOPDT (<i>First Order Plus Death-Time</i>)	12
2.5 Optimisasi dengan Metode <i>Genetic Algorithm</i>	13
BAB III METODOLOGI PENELITIAN	17
3.1 Pengumpulan Data	17
3.2 Pemodelan Proses <i>Regenerator Column</i> dengan Menggunakan HYSYS	20
3.3 Perhitungan <i>Gain Plant</i>	23
3.4 Optimisasi Pemisahan H ₂ S dengan Metode <i>Genetic Algorithm</i>	23
3.5 Pengujian Hasil Optimisasi	25

BAB IV HASIL DAN PEMBAHASAN	27
4.1 Pemodelan dan Validasi Proses <i>Regenerator</i> <i>Column</i>	27
4.2 Hasil Perhitungan <i>Gain Plant</i>	28
4.3 Hasil Perancangan Optimisasi dengan Metode <i>Genetic Algorithm</i>	29
4.4 Pengujian Hasil Optimisasi	32
4.5 Pembahasan.....	33
BAB V KESIMPULAN.....	35
DAFTAR PUSTAKA	37
LAMPIRAN	

DAFTAR GAMBAR

Gambar 2.1	Kondenser dan <i>reflux drum</i> dari kolom distilasi.....	6
Gambar 2.2	Kolom distilasi.....	7
Gambar 2.3	<i>Reboiler</i> dan <i>base column</i> dari kolom distilasi.....	7
Gambar 2.4	Kondisi di dalam <i>feed tray</i> (<i>tray</i> umpan).....	8
Gambar 2.5	Kondisi di dalam kolom distilasi	8
Gambar 2.6	Respon ketika diberikan input sinyal step	13
Gambar 2.7	Diagram blok optimisasi <i>genetic algorithm</i>	14
Gambar 2.8	Mekanisme <i>crossover</i>	16
Gambar 2.9	Mekanisme mutasi	16
Gambar 3.1	Diagram alir penelitian	17
Gambar 3.2	Simulasi <i>regenerator column</i> saat kondisi <i>steady state</i>	20
Gambar 3.3	<i>Process flow diagram regenerator column</i> saat kondisi dinamik dengan kontrolernya	22
Gambar 4.1	Temperatur pada <i>top column</i> saat laju aliran <i>hot oil reboiler</i> ditambah 1%	28
Gambar 4.2	Fraksi H_2S pada <i>top product</i> saat laju aliran <i>hot oil reboiler</i> ditambah 1%	29
Gambar 4.3	Hasil optimisasi saat 25 populasi dan 100 iterasi ...	30
Gambar 4.4	Hasil optimisasi saat 50 populasi dan 100 iterasi ...	30
Gambar 4.5	Hasil optimisasi saat 25 populasi dan 200 iterasi ...	31
Gambar 4.6	Hasil optimisasi saat 50 populasi dan 200 iterasi ...	31
Gambar 4.7	Perubahan temperatur pada <i>top column</i> setelah dilakukan optimisasi	32
Gambar 4.8	Perubahan nilai fraksi H_2S pada <i>top product</i> setelah dilakukan optimisasi	33

DAFTAR TABEL

Tabel 3.1	Dara Parameter <i>Feed Regenerator Column</i>	18
Tabel 3.2	Data Spesifikasi <i>Regenerator Column</i>	19
Tabel 3.3	Data Spesifikasi <i>Reboiler</i>	19
Tabel 3.4	Data Spesifikasi <i>Overhead Cooler</i>	19
Tabel 3.5	Data Spesifikasi <i>Reflux Drum</i>	19
Tabel 3.6	Data <i>Stream</i> dari <i>Regenerator Column</i>	21
Tabel 3.7	Data <i>Stream Hot Oil Reboiler</i> dan <i>Condenser</i>	21
Tabel 3.8	Rincian Kontrol pada <i>Regenerator Column</i>	22
Tabel 4.1	Kesetimbangan Massa dan Energi <i>Regenerator Column</i> dan Komponen.....	27
Tabel 4.2	Perbandingan Nilai Sebelum dan Sesudah Optimisasi	32
Tabel A.1	<i>Feed</i>	39
Tabel A.2	<i>Reflux</i>	40
Tabel A.3	<i>To Condenser</i>	41
Tabel A.4	<i>Top Product</i>	42
Tabel A.5	<i>To Reboiler</i>	43
Tabel A.6	<i>Bottom Product</i>	44
Tabel A.7	<i>Boil Up</i>	45
Tabel B. 1	Validasi Data Aliran <i>Feed</i>	47
Tabel B. 2	Validasi Data Aliran <i>Reflux</i>	48
Tabel B. 3	Validasi Data Aliran <i>To Condenser</i>	49
Tabel B. 4	Validasi Data Aliran <i>Top Product</i>	50
Tabel B. 5	Validasi Data Aliran <i>Bottom Product</i>	51

DAFTAR NOTASI

Notasi	Penjelasan	Besaran
F_B	Laju aliran <i>bottom</i>	kg/hr
F_D	Laju aliran distilat	kg/hr
F	Laju aliran <i>feed</i>	kg/hr
h_B	Entalpi cairan pada <i>bottom</i>	Joule/kg
h_D	Entalpi cairan pada distilat	Joule/kg
h_l	Entalpi <i>liquid</i>	Joule/kg
H_T	Entalpi <i>vapour</i> pada <i>top column</i>	Joule/kg
H_B	Entalpi <i>vapour</i> pada <i>boil up</i>	Joule/kg
h_1	Entalpi cairan pada <i>column</i> pertama	Joule/kg
K_p	<i>Gain sistem</i>	<i>dimensionless</i>
L	Laju aliran <i>liquid</i>	kg/hr
L_1	Laju aliran <i>liquid column</i> pertama	kg/hr
M_B	Total massa yang tertahan pada <i>reboiler</i>	kg
M_{RD}	Total massa yang tertahan pada distilat	kg
P	Tekanan (mutlak)	bar
P_c	Tekanan pada titik kritis	bar
Q_C	Kalor yang diberikan oleh <i>condenser</i>	kJ/hr
Q_R	Kalor yang diberikan oleh <i>reboiler</i>	kJ/hr
R	Konstantan gas ideal	8,3144 J/molK
T	Waktu	hr
T	Temperatur	°C
T_c	Temperatur pada titik kritis	°C
V_i	Volume molar	m ³ /mol

V_N	Laju aliran <i>vapour</i> pada <i>top column</i>	kg/hr
V	Laju aliran <i>vapour</i> pada <i>boil up</i>	kg/hr
X_B	Komposisi pada <i>bottom</i>	<i>dimensionless</i>
X_D	Komposisi pada distilat	<i>dimensionless</i>
X_f	Komposisi pada <i>feed</i>	<i>dimensionless</i>
X_I	Komposisi pada <i>column</i> pertama	<i>dimensionless</i>
Y_B	Komposisi pada <i>vapour</i> di <i>boil up</i>	<i>dimensionless</i>
Y_T	Komposisi pada <i>vapour</i> di <i>top column</i>	<i>dimensionless</i>
Z	Faktor <i>compressibility</i>	mol^{-1}
τ	<i>Time constant</i>	<i>second</i>
θ	<i>Death time</i>	<i>second</i>

BAB I PENDAHULUAN

1.1 Latar Belakang

Gas alam merupakan senyawa hidrokarbon (C_nH_{2n+2}) yang terdiri dari campuran beberapa macam gas hidrokarbon yang mudah terbakar dan non hidrokarbon (impuritas) seperti Hg, CO_2 , dan H_2S . Gas alam kemudian dikirim ke *Onshore Processing Facilities* (OPF) untuk dipisahkan dari fase *liquid*-nya, serta dihilangkan beberapa unsur impuritasnya sehingga diperoleh *sales gas*, yaitu gas yang siap untuk dijual (Haq, 2012).

Salah satu gas impuritas yang dihilangkan adalah H_2S . H_2S merupakan suatu gas tidak berwarna yang sangat beracun. Pada konsentrasi rendah, H_2S dapat tercium dengan ciri khas seperti bau telur busuk. Namun pada konsentrasi di atas 150 ppm H_2S sudah tidak tercium karena lumpuhnya indera penciuman manusia akibat menghirup H_2S terlalu banyak. Selain melumpuhkan indera penciuman, H_2S juga dapat menghilangkan kesadaran manusia bahkan kematian pada konsentrasi di atas 1000 ppm (Skrtic, 2006).

Dalam pengolahan gas alam terdapat proses menghilangkan kandungan *acid gas* dari *feed gas* pada *Acid Gas Removal Unit*. Pada unit tersebut mula-mula *feed gas* masuk ke dalam *absorber*. Kemudian *lean amine* akan masuk ke dalam *absorber* juga untuk menyerap kandungan *acid gas* dari *feed gas*. Keluaran dari *absorber* adalah *rich amine* pada *bottom product* dan *feed gas* tanpa kandungan *acid gas* pada *top product*. Selanjutnya *rich amine* menuju *regenerator column* untuk dipisahkan dari kandungan *acid gas* yang terserap tadi. Dari pemisahan tersebut dihasilkan *acid gas* yang berupa H_2S dan CO_2 serta *lean amine*. *Acid gas* yang telah dipisahkan akan di-*treatment* kembali di unit selanjutnya, sedangkan *lean amine* akan digunakan kembali pada proses di *absorber* (Haq, 2012).

Sunil Patil dan Viral Desai mengutarakan dalam penelitiannya bahwa kemurnian *top product* dan *bottom*

product dikendalikan oleh laju aliran *reflux* dan *hot oil reboiler* (Patil & Desai, 2009). Namun pada pemisahan H_2S di *regenerator column* ini hanya *hot oil reboiler* yang akan berpengaruh pada *top product*. Hal ini disebabkan karena *top product* keluar dari bagian atas *reflux drum*. Sedangkan *reflux* akan mempengaruhi *bottom product*. Sehingga untuk mendapatkan komposisi *top product* yang maksimal dapat dilakukan dengan merubah nilai laju aliran *hot oil reboiler*.

Terdapat berbagai metode dalam optimisasi, salah satunya yaitu menggunakan *genetic algorithm*. *Genetic algorithm* adalah teknik pencarian yang digunakan dalam komputasi untuk mencari solusi yang tepat atau perkiraan solusi untuk optimisasi dan masalah pencarian (Ce'sar, 2013). Pada teknik ini perlu memperhatikan beberapa parameter selain variabel yang ini dioptimisasi. Parameter yang perlu diperhatikan tersebut adalah jumlah populasi, jumlah generasi, jumlah bit, probabilitas *crossover*, probabilitas mutasi, dan *elitism*.

Dalam tugas akhir ini akan dilakukan optimisasi pemisahan H_2S di *regenerator column* melalui perubahan pada laju aliran *hot oil reboiler*. Pendekatan pada sistem ini menggunakan *first order plus death time* (FOPDT). Optimisasi ini akan dilakukan dengan metode *genetic algorithm*. Hasil dari optimisasi diharapkan dapat meningkatkan kualitas *top product* pada *regenerator column*.

1.2 Rumusan Masalah

Untuk mengoptimalkan pemisahan H_2S dapat dilakukan dengan merubah laju aliran *hot oil reboiler* menggunakan *gain* pada fungsi transfer. Kemudian metode optimisasi yang digunakan adalah *genetic algorithm*. Sehingga didapat rumusan masalah yaitu bagaimana mengoptimalkan pemisahan H_2S pada *regenerator column* tersebut dengan menggunakan *genetic algorithm*.

1.3 Tujuan

Adapun tujuan dari tugas akhir ini adalah mengoptimalkan pemisahan H_2S pada *regenerator column* dengan metode *genetic algorithm*.

1.4 Lingkup Kerja

Ruang lingkup dalam tugas akhir ini meliputi :

- *Regenerator column* yang digunakan sebagai model sistem dimodelkan dengan menggunakan HYSYS.
- Pendekatan pada sistem ini menggunakan *first order plus death time* (FOPDT).
- Optimisasi dilakukan dengan menggunakan metode *genetic algorithm*.
- Komposisi *top product* dikendalikan dengan memanipulasi laju aliran *hot oil reboiler*.

Halaman ini sengaja dikosongkan

BAB II DASAR TEORI

2.1 Pemurnian Gas

Pemurnian gas atau purifikasi gas merupakan sebuah proses yang berfungsi untuk mengurangi kandungan senyawa H_2S dan CO_2 pada suatu gas. Proses ini diperlukan karena unsur H_2S dan CO_2 bersifat korosif didalam air dan CO_2 sendiri merupakan senyawa yang bersifat tidak mudah terbakar, sehingga keberadaannya dapat menurunkan kualitas bahan bakar. Dalam pemrosesan gas, pemurnian gas biasanya terdapat pada awal pemrosesan gas atau apabila gas yang akan diolah masih dalam tekanan yang relatif rendah sehingga diperlukan kompresi maka proses pemurnian gas berada tepat setelah itu (Kidnay, Parrish, & McCartney, 2011).

Pemurnian gas dari kandungan H_2S dan CO_2 sendiri terbagi menjadi berbagai macam cara, seperti dengan menggunakan membran khusus, memberikan perlakuan fisik kepada gas ataupun dengan menggunakan senyawa kimiawi yang mampu menyerap kedua gas tersebut. Dalam tugas akhir kali ini tipe pemurnian gas yang akan digunakan adalah pemurnian gas dengan menggunakan senyawa kimiawi, yaitu *diethanol amine* (DEAmine).

Pemurnian gas dilakukan dengan cara mengontakan gas dengan DEAmine. Gas masuk dari bagian bawah *absorber/contactor* dan dikontakan secara langsung dengan DEAmine yang masuk dari bagian atas *contactor*, sehingga CO_2 dan H_2S yang terkandung pada gas diserap oleh DEAmine. Karena harga DEAmine sendiri yang relatif mahal dan DEAmine sendiri mampu dibersihkan untuk selanjutnya digunakan kembali maka pada proses ditambahkan sebuah unit *regenerator* yang mampu membersihkan DEAmine dengan memberikan perlakuan panas (Cumplings, Smith, & Nelsen, 2007).

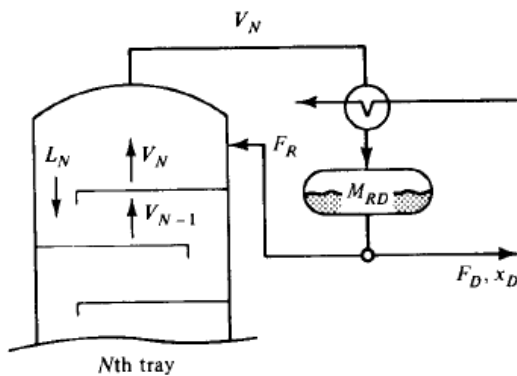
Maka dari itu pada proses pemurnian gas terdapat 2 buah subproses utama, yaitu proses penyerapan CO_2 dan H_2S dari gas ke DEAmine (*absorption*) dan pembersihan DEAmine

yang kaya akan CO_2 dan H_2S untuk dipakai kembali (*regeneration*) (Kazemi, Malayeri, & Shariati, 2014).

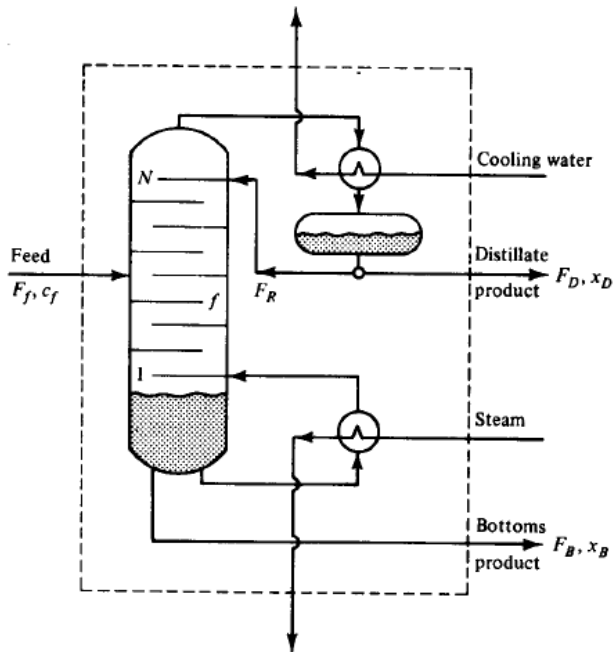
2.2 Regenerator Column

Regenerator column adalah kolom distilasi yang berfungsi untuk memisahkan gas dari fraksi berat lainnya. Tidak hanya itu, pemisahan komponen juga tergantung dari konsentrasi komponen tersebut. Dengan alasan inilah proses distilasi dikatakan tergantung pada karakteristik tekanan uap campuran. Fluida cair yang akan diproses dikenal sebagai *feed* dan dimasukkan ke nampan (*tray*) yang dinamakan *feed tray*. *Feed tray* membagi kolom menjadi bagian atas (*rectifying*) dan bagian bawah (*stripping*).

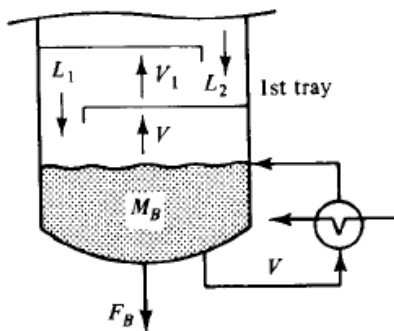
Di dalam kolom distilasi terdapat beberapa nampan (*trays*). Setiap *tray* memiliki 2 saluran di masing-masing sisinya yang disebut dengan *downcomers*. Fluida cair jatuh melalui *downcomers* dari satu *tray* ke *tray* lainnya (Stephanopoulos, 1984). *Tray* dirancang untuk memaksimalkan kontak antara fase gas dan cair dengan mempertimbangkan distribusi cairan dan distribusi uap. Hal ini dikarenakan bila kontak antara fase gas dan cair lebih maksimal maka akan lebih baik pemisahan yang terjadi pada setiap *tray*. Berikut merupakan gambaran sederhana dari kolom distilasi.



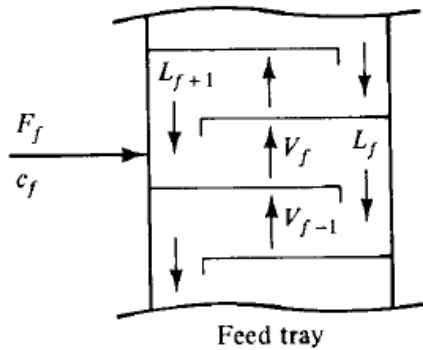
Gambar 2. 1 Kondenser dan *reflux drum* dari kolom distilasi (Stephanopoulos, 1984)



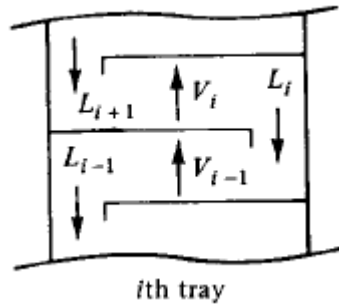
Gambar 2. 2 Kolom distilasi (Stephanopoulos, 1984)



Gambar 2. 3 Reboiler dan base column dari kolom distilasi (Stephanopoulos, 1984)



Gambar 2. 4 Kondisi di dalam *feed tray* (*tray* umpan)
(Stephanopoulos, 1984)



Gambar 2. 5 Kondisi di dalam kolom distilasi
(Stephanopoulos, 1984)

Tray pada kolom distilasi memiliki banyak lubang yang berfungsi untuk lajur aliran uap. Uap mengalir ke atas kolom dan dipaksa untuk melewati cairan melalui bukaan pada setiap *tray*. Saat uap panas melewati cairan dari *tray* satu ke *tray* lainnya, uap tersebut mentransfer panas ke cairan sehingga sebagian uap terkondensasi dan menambah cairan pada *tray*. Sisa uap lainnya kemudian keluar melalui atas kolom dan didinginkan oleh *condenser*. Sedangkan cairan ini dimasukkan kembali ke atas kolom dan disebut *reflaks*. Panas dipasok ke *reboiler* untuk menghasilkan uap. Uap pada *reboiler* tersebut

kembali dimasukkan ke dalam unit melalui bagian bawah kolom. Cairan keluaran dari *reboiler* dikenal sebagai *bottom product* (Luyben, 1992).

Kesetimbangan massa total dari kolom distilasi adalah sebagai berikut:

$$F_f = F_D + F_B \quad (2.1)$$

dengan kesetimbangan massa komponen kolom distilasi adalah sebagai berikut:

$$F_f X_f = F_D X_D + F_B X_B \quad (2.2)$$

kemudian untuk kesetimbangan energi pada kolom distilasi adalah sebagai berikut:

$$F_f h_f - F_D h_D + F_B h_B + Q_R - Q_C = 0 \quad (2.3)$$

Sedangkan pada kondisi dinamik, kesetimbangan kolom distilasi dapat dinyatakan dalam persamaan sebagai berikut :

- a. Kesetimbangan massa pada kondensor dan *reflux drum*.
Neraca massa total:

$$\frac{dM_{RD}}{dt} = V_N - F_R - F_D \quad (2.4)$$

Neraca massa komponen:

$$\frac{d(M_{RD} X_D)}{dt} = V_N Y_N - (F_R + F_D) X_D \quad (2.5)$$

Neraca massa energi:

$$\frac{d(M_{RD} h_D)}{dt} = V_N H_N - F_R h_R - F_D h_D - Q_C \quad (2.6)$$

- b. Kesetimbangan massa pada setiap *tray*
Neraca massa total:

$$\frac{dM_i}{dt} = L_{i+1} - L_i + V_{i-1} - V_i \quad (2.7)$$

Neraca massa komponen:

$$\frac{dM_i X_i}{dt} = L_{i+1} X_{i+1} - L_i X_i + V_{i-1} Y_{i-1} - V_i Y_i \quad (2.8)$$

Neraca massa energi:

$$\frac{dM_i h_i}{dt} = L_{i+1} h_{i+1} - L_i h_i + V_{i-1} H_{i-1} - V_i H_i \quad (2.9)$$

- c. Kesetimbangan massa pada *tray* umpan
Neraca massa total:

$$\frac{dM_f}{dt} = F_f + L_{f+1} - L_f + V_{f-1} - V_f \quad (2.10)$$

Neraca massa komponen:

$$\begin{aligned} \frac{dM_f X_f}{dt} = & F_f X_f + L_{f+1} X_{f+1} - L_f X_f + \\ & V_{f-1} Y_{f-1} - V_f Y_f \end{aligned} \quad (2.11)$$

Neraca massa energi:

$$\begin{aligned} \frac{dM_f h_f}{dt} = & F_f h_f + L_{f+1} h_{f+1} - L_f h_f + \\ & V_{f-1} H_{f-1} - V_f H_f \end{aligned} \quad (2.12)$$

- d. Kesetimbangan massa pada *reboiler* dan *base column*
Neraca massa total:

$$\frac{dM_B}{dt} = L_1 - V - F_B \quad (2.13)$$

Neraca massa komponen:

$$\frac{d(M_B X_B)}{dt} = L_1 X_1 - V Y_B - F_B X_B \quad (2.14)$$

Neraca massa energi:

$$\frac{d(M_B h_B)}{dt} = L_1 h_1 - V H_B - F_B h_B + Q_R \quad (2.15)$$

2.3 Peng-Robinson Equation of State

Equation of state (persamaan keadaan) adalah persamaan termodinamika yang menggambarkan keadaan materi di bawah keadaan fisik yang ada. Persamaan ini merupakan persamaan konstitutif yang menyediakan hubungan matematis antara dua atau lebih fungsi keadaan terkait dengan zat atau unsur, seperti suhu, tekanan, volume, atau energi internal. Persamaan keadaan berguna untuk mengkorelasikan kepadatan gas dan cairan dengan suhu dan tekanan.

Terdapat beberapa macam *equation of state* salah satunya yaitu Peng-Robinson *equation of state* yang dinyatakan dalam rumus di bawah ini.

$$Z_i = \frac{PV_i}{RT} = \frac{V_i}{V_i - b_i} - \frac{\alpha_i(T)V_i}{RT(V_i + \varepsilon b_i)(V_i + \sigma b_i)} \quad (2.16)$$

$$a_i = 0,45724 \frac{R^2 T_c^2}{P_c} \quad (2.17)$$

$$b_i = 0,07780 \frac{RT_c}{P_c} \quad (2.18)$$

$$k = 0,37464 + 1,5422\omega - 0,26922\omega^2 \quad (2.19)$$

$$\alpha_i = \left(1 + \kappa(1 - T_r^{0,5})\right)^2 \quad (2.20)$$

$$T_r = \frac{T}{T_c} \quad (2.21)$$

Persamaan Peng-Robinson ditemukan pada tahun 1976 guna menyempurnakan persamaan sebelumnya di *University of Alberta* oleh Ding-Yu Peng dan Donald Robinson untuk memenuhi tujuan berikut (Peng & Robinson, 1976):

- Parameter-parameter dapat dinyatakan dalam sifat kritis dan faktor aksentrik.
- Model memberikan tingkat akurasi yang mendekati titik kritis, terutama untuk perhitungan faktor kompresibilitas dan densitas cairan.
- Aturan dalam pencampuran tidak harus menggunakan lebih dari satu parameter interaksi biner tunggal, yang harus independen dari tekanan suhu dan komposisi.
- Persamaan berlaku untuk semua perhitungan dari semua sifat fluida dalam proses gas alam.

2.4 Pendekatan FOPDT (*First Order Plus Death-Time*)

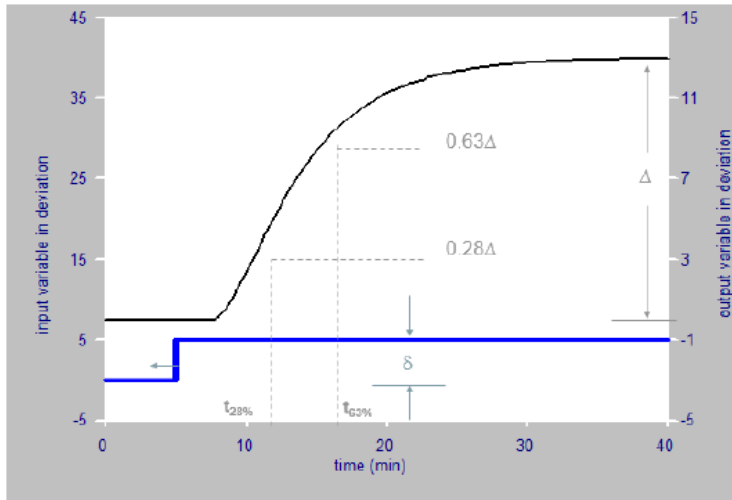
Fungsi transfer FOPDT (*First Order Plus Death-Time*) dapat digunakan untuk memodelkan sistem dinamik dari suatu proses industri sebagai berikut:

$$G(s) = \frac{K_p e^{-\theta s}}{\tau s + 1} \quad (2.22)$$

K_p merupakan *gain* sistem yang diperoleh dari perbandingan perubahan proses variabel yang diukur dan sinyal output dari *controller* dengan rumus matematis sebagai berikut.

$$K_p = \frac{\Delta \text{Measured Process Variable}}{\Delta \text{Manipulated Variable}} \quad (2.23)$$

τ (*time constant*) adalah waktu yang dibutuhkan oleh proses variabel untuk mencapai 63,2 % dari perubahan total akhir. Sedangkan θ (*dead time*) adalah perbedaan waktu dari respon *output controller* yang berupa sinyal step dan respon proses variabel.



Gambar 2. 6 Respon ketika diberikan input sinyal step (Marlin, 2000)

2.5 Optimisasi dengan Metode *Genetic Algorithm*

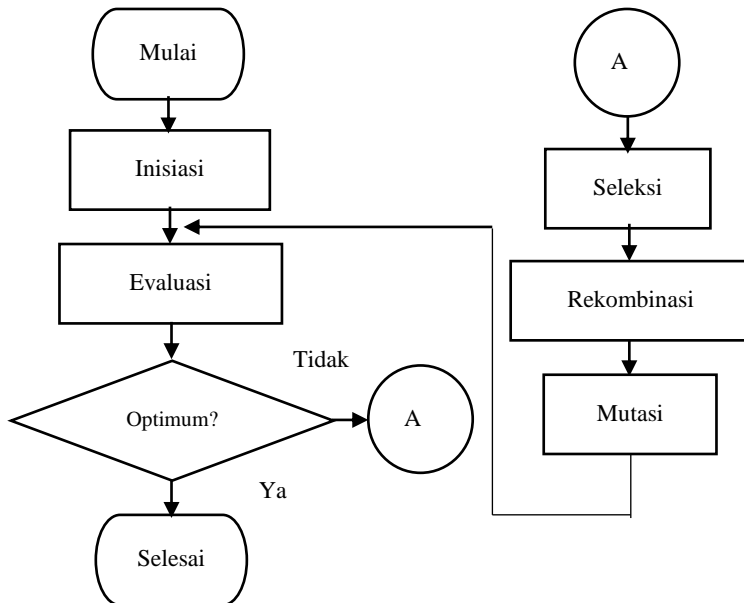
Secara umum *genetic algorithm* merupakan teknik pencarian yang digunakan dalam komputasi untuk mencari solusi yang tepat atau perkiraan solusi untuk optimisasi dan masalah pencarian (Ce'sar, 2013). Suatu *genetic algorithm* standar membutuhkan dua hal untuk didefinisikan, yaitu sebuah *genetic representation* dari sebuah *solution domain* (domain solusi) dan sebuah *fitness function* untuk mengevaluasi sebuah domain solusi. Representasi standar dari solusinya adalah sebuah *array of bits* (larik bit). Properti utama yang membuat representasi genetik ini baik adalah bagian-bagiannya yang bisa diakses dengan mudah karena ukuran yang pasti (*fixed*), yang memudahkan suatu operasi persilangan yang sederhana. Representasi panjang variabel juga digunakan disini, tetapi implementasi persilangan jauh lebih sulit pada kasus ini.

Fungsi penghitung nilai kecocokan (*fitness*) didefinisikan pada representasi *genetic* dan digunakan untuk mengukur kualitas (*quality*) pada solusi yang direpresentasikan. Fungsi

penghitung ini selalu tergantung pada masalah yang ada (*problem dependent*).

Setelah memiliki representasi genetik dan sebuah fungsi untuk mencari nilai kecocokan (*fitness*) terdefinisi, maka *genetic algorithm* akan melanjutkan untuk membentuk suatu populasi acak, kemudian meningkatkannya melalui aplikasi yang berulang-ulang dari mutasi, persilangan, dan operator seleksi.

Genetic algorithm dapat dituliskan dalam berbagai bahasa pemrograman. Namun tahapan logika yang digunakan dalam GA adalah sama. Diagram alir optimisasi menggunakan GA dapat dilihat pada Gambar 2.7.

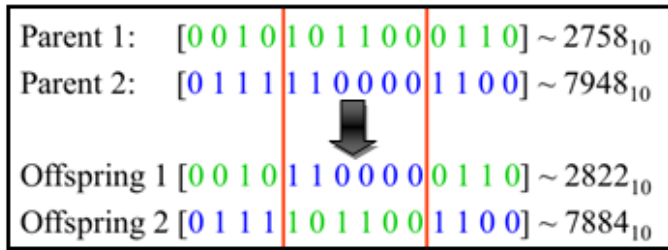


Gambar 2. 7 Diagram blok optimisasi *genetic algorithm*

Tahapan-tahapan *genetic algorithm* diantaranya yaitu:

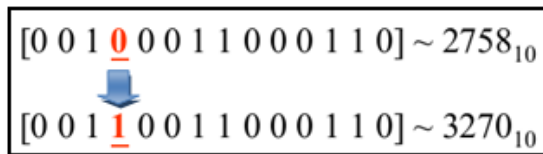
- Inisiasi
Populasi awal dari kandidat solusi biasanya dicari secara acak dalam seluruh ruang pencarian.

- **Pengkodean Kromosom**
Kromosom pada GA merupakan solusi dari satu variabel. Jenis kode yang digunakan adalah biner, yaitu 0 atau 1. Kromosom diwakili oleh beberapa gen. Kromosom dalam bentuk biner merupakan kromosom genotip dan yang berupa solusi adalah kromosom fenotip.
- **Evaluasi**
Ketika populasi yang telah diinisiasi atau populasi keturunan terbentuk, nilai *fitness* dari tiap individu dievaluasi. Nilai *fitness* merupakan nilai dari kemampuan solusi untuk bertahan.
- **Seleksi**
Seleksi mengalokasikan lebih banyak salinan dari solusi dengan *fitness* yang lebih tinggi dan memberlakukan mekanisme survival dari tiap *fitness* pada tiap kandidat solusi. Gagasan utama dari tahapan seleksi adalah untuk mendapatkan solusi terbaik dari generasi terburuk, dan banyak prosedur seleksi telah ditemukan, seperti *roulette-wheel*, *selection stochastic universal*, seleksi *ranking*, seleksi turnamen, dan lain sebagainya.
- **Rekombinasi**
Tahap rekombinasi mengkombinasi bagian dari dua atau lebih solusi induk untuk membentuk individu baru dengan kemungkinan menjadi solusi yang lebih baik. Biasanya rekombinasi menggunakan mekanisme *crossover*. Pada Gambar 2.8 merupakan mekanisme *crossover* dimana pada kromosom induk dipotong oleh *crossover point* sehingga gen-gen pada tiap kromosom bertukar silang dan menghasilkan anak.



Gambar 2. 8 Mekanisme *crossover* (Ce'sar, 2013)

- **Mutasi**
Ketika rekombinasi beroperasi terhadap dua atau lebih kromosom, mutasi lokal tetapi acak memodifikasi sebuah solusi. Dan juga akan terjadi bermacam-macam mutasi, tetapi biasanya melibatkan satu atau lebih perubahan sifat individu. Mekanisme mutasi dapat dilihat pada Gambar 2.9, salah satu gen pada kromosom awal dimutasi sehingga menghasilkan kromosom baru.

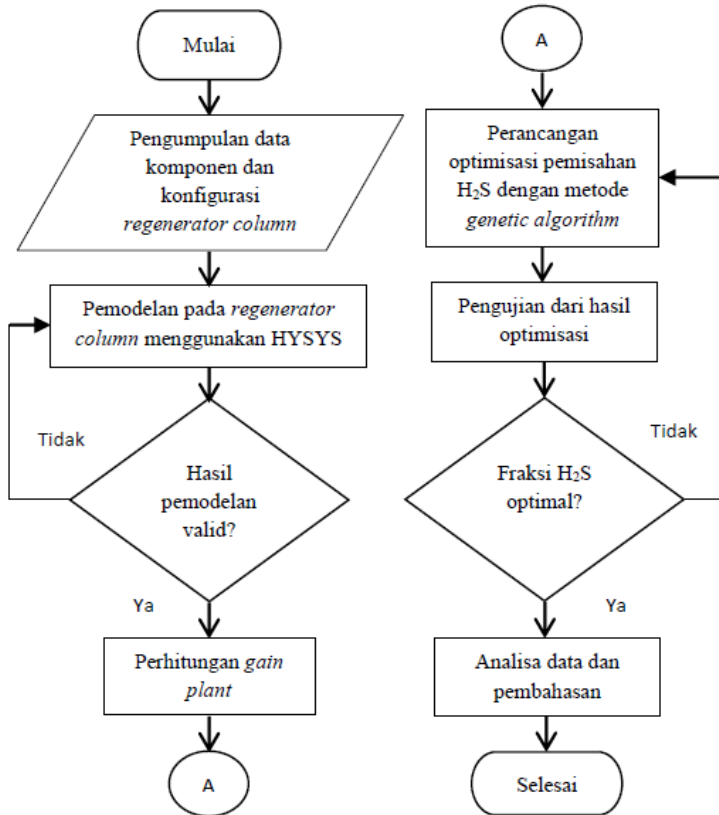


Gambar 2. 9 Mekanisme mutasi (Ce'sar, 2013)

- **Penggantian**
Keturunan hasil dari seleksi, rekombinasi, dan mutasi akan menggantikan populasi induk. Banyak metode penggantian seperti penggantian *elitism*, penggantian *generation-wise*, dan penggantian *steady-state*.

BAB III METODOLOGI PENELITIAN

Langkah-langkah yang dilakukan pada penelitian ini dapat dijelaskan melalui diagram alur berikut.



Gambar 3. 1 Diagram alir penelitian

3.1 Pengumpulan Data

Data yang dikumpulkan untuk tugas akhir ini berupa parameter dan data aliran *feed regenerator column*. Data tersebut kemudian digunakan untuk simulasi sistem. Untuk

data temperatur, tekanan, dan laju aliran massa masing-masing adalah 93,54 °C, 5,02 bar, dan 49494,94 kg/hr. Data parameter *feed regenerator column* dapat dilihat pada tabel berikut ini.

Tabel 3.1 Parameter *Feed Regenerator Column*

No.	Nama Komponen	Komposisi
1.	H ₂ S	0,001967
2.	CO ₂	0,030249
3.	Nitrogen	0
4.	Methane	0,000160
5.	Ethane	0,000027
6.	Propane	0,000018
7.	i-Butane	0,000001
8.	n-Butane	0,000001
9.	i-Pentane	0
10.	n-Pentane	0
11.	n-Hexane	0
12.	n-Heptane	0
13.	n-Octane	0
14.	n-Nonane	0
15.	n-Decane	0
16.	n-C ₁₁	0
17.	H ₂ O	0,868232
18.	DEAmine	0,099343

Kemudian dibutuhkan pula data spesifikasi alat yang digunakan. Data tersebut juga akan digunakan untuk simulasi sistem. Data spesifikasi yang digunakan adalah sebagai berikut.

Tabel 3.2 Data Spesifikasi *Regenerator Column*

No.	Spesifikasi	Nilai
1.	<i>Size</i>	1500 x 22100 mm
2.	<i>Diameter</i>	1,5 m
3.	<i>Design Pressure</i>	3,5 barg
4.	<i>Design Temperature</i>	200 / 0 °C
5.	<i>Number of Tray</i>	20
6.	<i>Internal Tray Type</i>	Valve

Tabel 3.3 Data Spesifikasi *Reboiler*

No.	Spesifikasi	Nilai	
		<i>Shell</i>	<i>Tube</i>
1.	<i>Design Pressure</i>	28 bar	3,5 bar
2.	<i>Design Temperature</i>	250 / 0 °C	00 / 0 °C
3.	<i>Type</i>	<i>Vertical Thermosiphon</i>	
4.	<i>Design Duty</i>	3889 kW	

Tabel 3.4 Data Spesifikasi *Overhead Cooler*

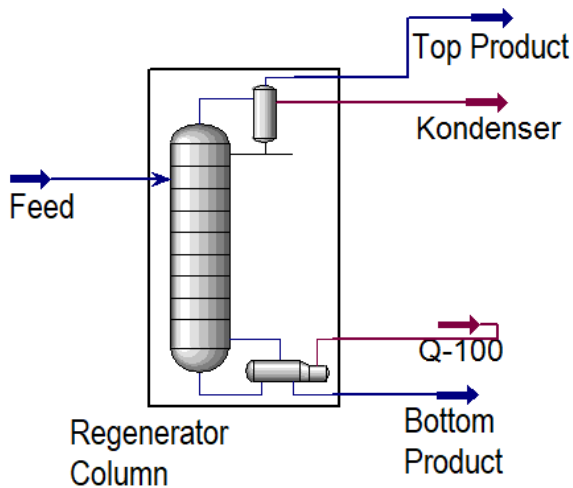
No.	Spesifikasi	Nilai
1.	<i>Type</i>	<i>Air Cooler</i>
2.	<i>Inlet / Outlet</i>	94,7 / 48,9 °C
3.	<i>Design Temperature</i>	200 / 0 °C
4.	<i>Design Pressure</i>	3,5 barg
5.	<i>Duty</i>	1066,3 kW

Tabel 3.5 Data Spesifikasi *Reflux Drum*

No.	Spesifikasi	Nilai
1.	<i>Size</i>	2000 x 4000 mm
2.	<i>Design Pressure</i>	3,5 barg
3.	<i>Design Temperature</i>	200 / 0 °C

3.2 Pemodelan Proses *Regenerator Column* dengan Menggunakan HYSYS

Pemodelan pada proses *regenerator column* dilakukan menggunakan HYSYS. Pada pemodelan ini dilakukan simulasi berdasarkan data spesifikasi dan *feed regenerator column* yang telah didapat untuk mendapatkan kondisi *real plant*. Berikut ini merupakan tampilan simulasi *regenerator column* saat kondisi *steady state*.



Gambar 3. 2 Simulasi *regenerator column* saat kondisi *steady state*

Dimana *feed* merupakan masukan dari *regenerator column*. Kemudian Q-100 merupakan *heat flow* dari *reboiler*. Berikut merupakan data masing-masing *stream* dari *regenerator column*.

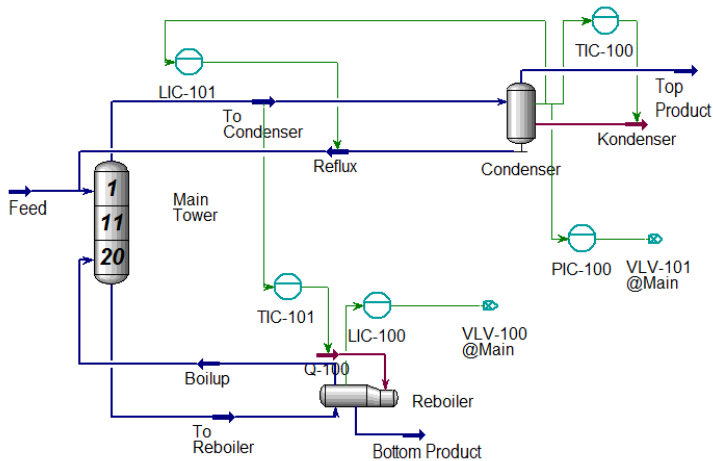
Tabel 3.6 Data Stream dari *Regenerator Column*

No.	Stream	Molar Flow Rate (kgmole/hr)	Fraction	Enthalpy (kJ/kgmole)
1.	Feed	1801	0,002	-305000
2.	Reflux	69,53	0	-279900
3.	To condenser	274,9	0,0129	-265600
4.	Top product	205,4	0,01725	-274800
5.	To reboiler	2045	0	-294900
6.	Boil up	449,6	0	-238900
7.	Bottom product	1595	0	-299300

Tabel 3.7 Data Stream Hot Oil Reboiler dan Condenser

No.	Stream	Heat flow (kJ/hr)
1.	Hot oil	18162200
2.	Condenser	2891927

Sehingga kesetimbangan massa dan energi dari *regenerator column* dapat diperoleh dengan menggunakan persamaan 2.1 sampai dengan 2.15. Selanjutnya *regenerator column* diubah menjadi keadaan dinamik dengan cara dilakukan *sizing* dan pemasangan kontroler dengan memperhatikan data yang dikumpulkan sebelumnya.. Berikut ini merupakan tampilan simulasi *regenerator column* pada kondisi dinamik dan penjelasan mengenai kontroler yang terpasang.



Gambar 3.3 *Process flow diagram regenerator column saat kondisi dinamik dengan kontrolernya*

Tabel 3.8 Rincian Kontrol pada *Regenerator Column*

No.	Kontroler	<i>Manipulated Variable</i>	<i>Process Variable</i>
1.	PIC-100	Laju aliran <i>top product</i>	Tekanan pada <i>reflux drum</i>
2.	TIC-100	Laju <i>condenser</i>	Temperatur pada <i>reflux drum</i>
3.	TIC-101	Laju aliran <i>hot oil reboiler</i>	Temperatur pada <i>top column</i>
4.	LIC-100	Laju aliran <i>bottom product</i>	Level pada <i>reboiler</i>
5.	LIC-101	Laju aliran <i>reflux</i>	Level pada <i>reflux drum</i>

Laju aliran *hot oil reboiler* sebelum optimisasi adalah 18162200 kJ/hr dengan temperatur pada *top column* sebesar 109,22 °C. Sedangkan nilai fraksi H₂S pada *top product* sebelum optimisasi sebesar 0,017246.

3.3 Perhitungan Gain Plant

Pemodelan yang telah dilakukan dapat dilihat pada gambar 3.3. Dari pemodelan tersebut selanjutnya dilakukan uji *open loop* dengan memberikan bukaan tambahan pada *manipulated variable* sebanyak 1%. Kemudian diamati pengaruh dari uji *open loop* tersebut terhadap *process variable*. Pengamatan ini dilakukan hingga respon dari sistem sudah stabil sehingga didapatkan nilai *gain plant*.

Nilai *gain plant* yang telah didapatkan dari uji *open loop* tersebut selanjutnya dimasukkan ke *genetic algorithm*. Nilai *gain plant* diperoleh dengan menggunakan rumus dibawah ini:

$$Gain_{plant} = \frac{\Delta Process Variable}{\Delta Manipulated Variable} \quad (3.1)$$

Pada penelitian ini *process variable* merupakan nilai X_D dimana X_D merupakan fraksi H_2S pada *top product*. Sedangkan *manipulated variable* merupakan nilai Q_r atau laju aliran *hot oil reboiler*. X_D dan Q_R awal merupakan ketika kontroler laju aliran *hot oil reboiler* belum diberi bukaan tambahan. Kemudian X_D dan Q_R akhir merupakan ketika kontroler tersebut diberi bukaan tambahan 1%. Maka saat laju aliran *hot oil reboiler* diubah, *gain plant* yang didapat adalah:

$$\begin{aligned} X_{D(awal)} &= 0,017246 \\ X_{D(akhir)} &= 0,016788 \\ Q_{R(awal)} &= 18162200 \text{ kJ/hr} \\ Q_{R(akhir)} &= 19158100 \text{ kJ/hr} \end{aligned}$$

$$G_1 = \frac{0.016788 - 0.017246}{19158100 - 18162200} = -00000459 \quad (3.2)$$

3.4 Optimisasi Pemisahan H_2S dengan Metode Genetic Algorithm

Untuk menentukan nilai optimal dari konsentrasi H_2S yang terpisah, digunakan optimisasi dengan metode *genetic algorithm* menggunakan *software* MatLab. Variabel yang akan

diubah adalah laju aliran *hot oil reboiler*. *Constrain* dari variabel tersebut berdasarkan batas minimal dan maksimal laju aliran *hot oil*.

Nilai *gain plant* yang telah diperoleh dari respon sistem kemudian digunakan untuk optimisasi. Nilai *gain plant* ini digunakan untuk optimisasi karena optimisasi ini dilakukan untuk mendapatkan *set point* temperatur *top column* terbaik. Pada *genetic algorithm* terdapat *objective function* yang merupakan nilai yang harus dioptimalkan. Pada penelitian ini nilai *objective function* yang berupa jumlah *error* X_D harus diminimalkan dengan nilai fraksi yang berupa nilai persen. Nilai *error* X_D dapat diperoleh dari rumus berikut:

$$\Delta X_{target} = X_{target} - X_{awal} \quad (3.3)$$

$$\Delta X_{actual} = X_{optimisasi} - X_{awal} \quad (3.4)$$

$$error\ X = \Delta X_{target} - \Delta X_{actual} \quad (3.5)$$

Optimisasi dengan metode *genetic algorithm* ini perlu memperhatikan beberapa parameter selain laju aliran *hot oil reboiler*. Jumlah populasi dari optimisasi ini akan dilakukan dengan populasi sebanyak 25 dan 50. Untuk jumlah generasi adalah 100 dan 200. Jumlah bit yang digunakan adalah 15 bit yang akan membentuk kromosom secara acak untuk menghasilkan sisa konsentrasi H_2S yang minimal. Nilai probabilitas *cross over* dan mutasi yang digunakan masing-masing sebesar 70% dan 10%. Hal ini berarti bahwa 70% dari kromosom yang memiliki keturunan yang baik akan dikawinkan kembali agar menghasilkan keturunan yang lebih baik. Sedangkan probabilitas mutasi yang bernilai 10% berarti bahwa 10% hasil dari *cross over* secara acak akan dibalik nilainya. Kemudian untuk nilai dari *elitism* adalah 80%. Hal ini berarti bahwa 80% dari hasil yang paling baik akan dibawa ke generasi selanjutnya sebagai generasi induk.

3.5 Pengujian Hasil Optimisasi

Setelah didapatkan nilai laju aliran *hot oil reboiler* dari *genetic algorithm*, selanjutnya nilai tersebut diuji. Pengujian ini dilakukan dengan memasukkan nilai tersebut pada pemodelan yang telah dibuat. Kemudian dilakukan perbandingan nilai pemisahan H_2S setelah dioptimisasi dengan sebelum dioptimisasi. Apabila hasil yang didapat dari *genetic algorithm* belum optimal, maka dilakukan perancangan ulang untuk menemukan hasil yang paling optimal. Jika hasil yang didapat dari *genetic algorithm* sudah lebih baik dari sebelumnya, maka dapat dikatakan bahwa *genetic algorithm* telah berhasil melakukan optimisasi.

Halaman ini sengaja dikosongkan

BAB IV HASIL DAN PEMBAHASAN

4.1 Pemodelan dan Validasi Proses *Regenerator Column*

Pemodelan yang dilakukan dapat dilihat pada gambar 3.3. Hasil perhitungan *mass energy balance* pada tabel 4.1 menunjukkan bahwa *error* dari simulasi sangat kecil. *Error* dari masing-masing kesetimbangan seluruhnya di bawah 1%. Hal ini mengindikasikan bahwa model dari *regenerator column* sudah benar.

Pada tabel 4.1, *input* dari kolom adalah aliran *feed*, *reflux*, dan *boil up*. *Output* dari kolom adalah aliran *to condenser* dan *to reboiler*. Untuk *input* dari kondenser dan *reboiler* masing-masing adalah *to condenser* dan *to reboiler*. *Output* dari kondenser adalah *reflux* dan *top product*, sedangkan *output* dari *reboiler* adalah *boil up* dan *bottom product*.

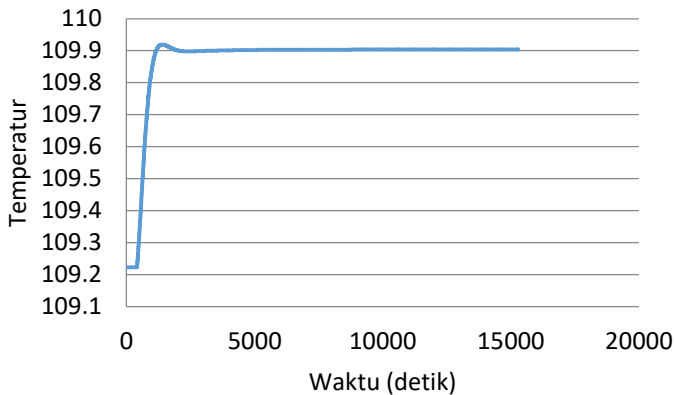
Tabel 4.1 Kesetimbangan Massa dan Energi *Regenerator Column* dan Komponen

Kesetimbangan		Kolom	Kondenser	<i>Reboiler</i>
<i>Mass Balance</i> (kg/hr)	<i>Input</i>	58861,94	6410	52451,80
	<i>Output</i>	58860,00	6410	52451,86
	<i>Error</i>	1,94	0	-0,06
	<i>%Error</i>	0	0	0
<i>Heat Balance</i> (kJ/hr)	<i>Input</i>	-676060000	-76300000	-584840000
	<i>Output</i>	-676010000	-75900000	-584900000
	<i>Error</i>	0	-409000	0
	<i>%Error</i>	0	0,54	0
<i>Component Balance</i> (kgmole/hr)	<i>Input</i>	3,6	3,56	0,00063
	<i>Output</i>	3,57	3,54	0,00064
	<i>Error</i>	0,03	0,02	0
	<i>%Error</i>	0,83	0,6	0

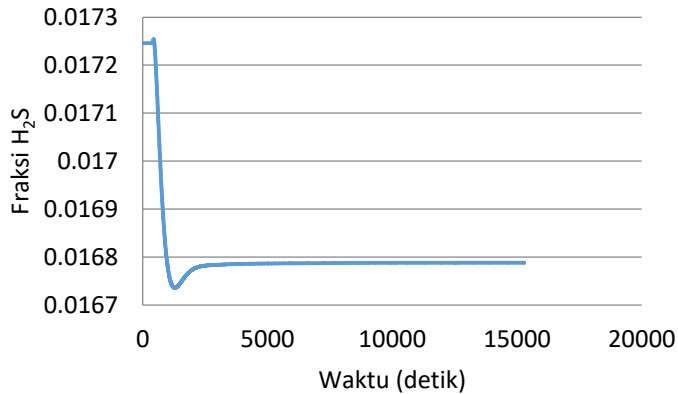
Setelah dilakukan perhitungan *mass energy balance*, selanjutnya dilakukan validasi. Seperti yang terdapat pada lampiran B, *error* pada masing-masing aliran *regenerator column* rata-rata dibawah 5%. Nilai *error* yang cukup kecil ini mengindikasikan bahwa model dari *regenerator column* cukup valid.

4.2 Hasil Perhitungan *Gain Plant*

Setelah kontroler *hot oil reboiler* diberikan bukaan tambahan 1% dari bukaan awal kemudian dilakukan pengambilan data respon sistemnya. Laju aliran *hot oil reboiler* setelah diberikan bukaan tambahan mengalami perubahan menjadi 19158100 kJ/hr. Gambar 4.1 dan 4.2 berikut ini merupakan grafik dari respon sistem tersebut.



Gambar 4. 1 Temperatur pada *top column* saat laju aliran *hot oil reboiler* ditambah 1%

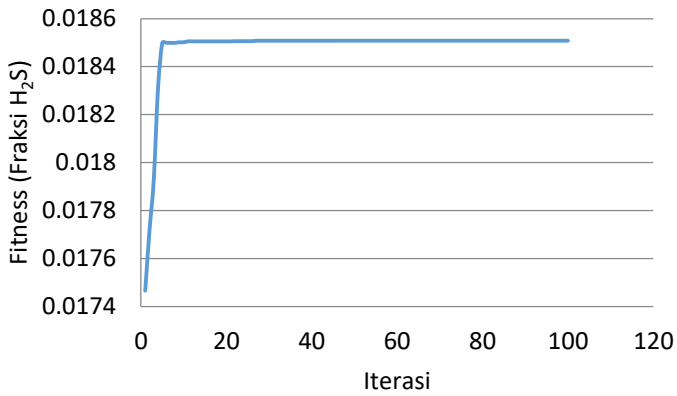


Gambar 4. 2 Fraksi H_2S pada *top product* saat laju aliran *hot oil reboiler* ditambah 1%

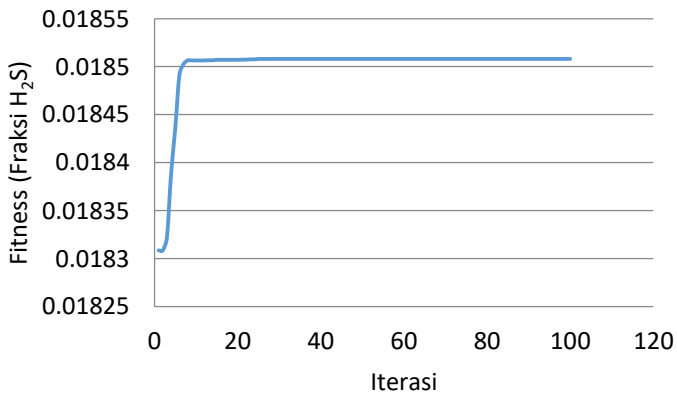
Pada gambar 4.1 terlihat bahwa temperatur pada *top column* meningkat menjadi 109,9 °C. Kemudian pada gambar 4.2 terlihat bahwa fraksi H_2S mengalami penurunan hingga mencapai 0,016788.

4.3 Hasil perancangan Optimisasi dengan Metode *Genetic Algorithm*

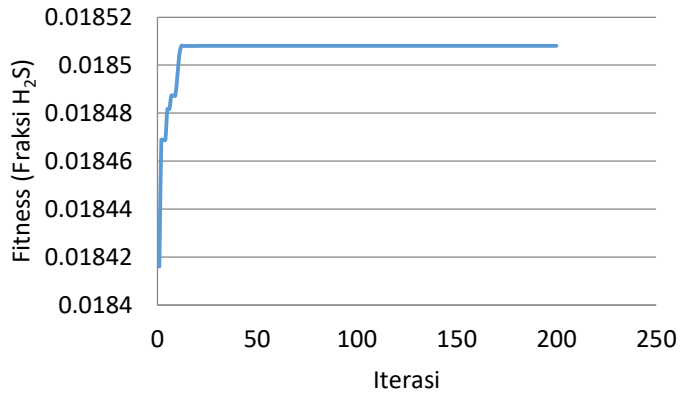
Optimisasi ini dilakukan untuk mengoptimalkan pemisahan H_2S pada *regenerator column* dengan mengubah variabel yang dioptimisasi. Variabel yang dioptimisasi adalah nilai laju aliran *hot oil reboiler*. *Constrain* yang digunakan pada optimisasi ini adalah laju aliran *hot oil reboiler* di antara 15216703-19158100 kJ/hr. Variasi jumlah populasi adalah 25 dan 50, sedangkan variasi jumlah iterasi adalah 100 dan 200. Kemudian nilai X_D target yang digunakan sebesar 0,0186. Berikut ini merupakan hasil dari optimisasi tersebut.



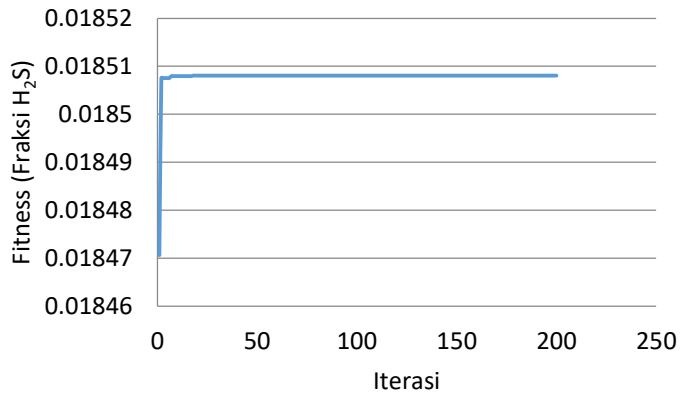
Gambar 4. 3 Hasil optimisasi saat 25 populasi dan 100 iterasi



Gambar 4. 4 Hasil optimisasi saat 50 populasi dan 100 iterasi



Gambar 4. 5 Hasil optimisasi saat 25 populasi dan 200 iterasi



Gambar 4. 6 Hasil optimisasi saat 50 populasi dan 200 iterasi

Nilai terbaik laju aliran *hot oil reboiler* pada optimisasi ini adalah 15216823 kJ/hr dengan fraksi H_2S pada *top product* sebesar 0,018508. Pada gambar 4.3-4.6 didapatkan hasil terbaik masing-masing pada iterasi ke 47, 29, 21 dan 18.

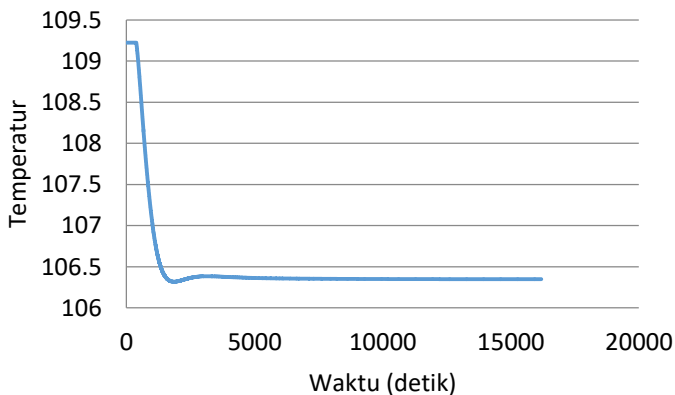
4.4 Pengujian Hasil Optimisasi

Setelah didapatkan hasil optimisasi selanjutnya dilakukan pengujian hasil tersebut. Pengujian ini dilakukan dengan melakukan simulasi menggunakan hasil dari optimisasi. Berikut ini merupakan rincian hasil simulasi dari proses *regenerator column*.

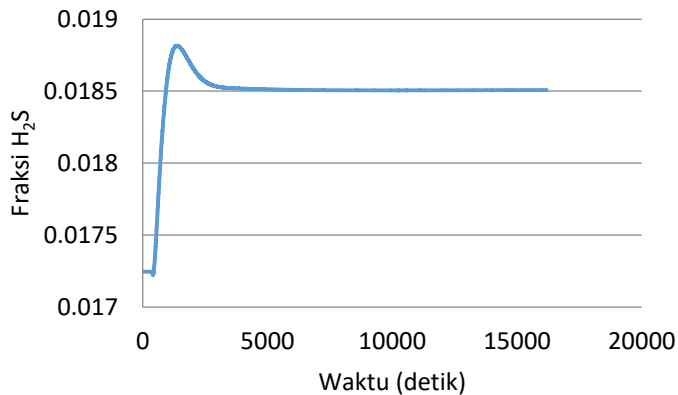
Tabel 4.2 Perbandingan Nilai Sebelum dan Sesudah Optimisasi

No.	Parameter	Sebelum Optimisasi	Sesudah Optimisasi
1.	X_D	0,017246	0,018508
2.	Temperatur	109,22 °C	106,35 °C
3.	Hot oil	18162200 kJ/hr	15216823 kJ/hr

Dari tabel 4.2 dapat dilihat bahwa temperatur pada *top column* mengalami penurunan menjadi 106,35 °C. Kemudian fraksi H_2S mengalami peningkatan menjadi 0,018508. Berikut ini merupakan grafik pengujian hasil optimisasi,



Gambar 4. 7 Perubahan temperatur pada *top column* setelah dilakukan optimisasi



Gambar 4.8 Perubahan nilai fraksi H_2S pada *top product* setelah dilakukan optimisasi

Dari gambar 4.7 terlihat bahwa temperatur pada *top column* mengalami penurunan hingga mencapai 106,35 °C. Kemudian pada gambar 4.8 dapat dilihat bahwa fraksi H_2S mengalami peningkatan hingga mencapai 0,018508. Nilai fraksi H_2S yang lebih baik ini menunjukkan bahwa *genetic algorithm* berhasil melakukan optimisasi dengan menemukan nilai laju aliran *hot oil reboiler* yang optimal.

4.5 Pembahasan

Pemodelan *regenerator column* telah dilakukan menggunakan HYSYS. Hasil perhitungan *mass energy balance* yang didapatkan dari pemodelan tersebut menghasilkan nilai *error* yang sangat kecil. Nilai *error* dari masing-masing kesetimbangan seluruhnya di bawah 1%. Hal ini menunjukkan bahwa hasil pemodelan tersebut sudah benar. Kemudian nilai *error* pada masing-masing aliran *regenerator column* rata-rata dibawah 5%. Nilai *error* yang cukup kecil ini mengindikasikan bahwa model dari *regenerator column* sudah cukup valid.

Kualitas *top product* dapat dilihat dari fraksi masing-masing komponen pada *top product*. Pada perhitungan *gain*

plant, dilakukan uji *open loop* dengan menambah bukaan pada laju aliran *hot oil reboiler* sebesar 1%. Sehingga laju aliran *hot oil reboiler* menjadi 19158100 kJ/hr. Temperatur pada *top column* yang dipengaruhi oleh laju aliran *hot oil reboiler* juga berubah menjadi 109,9 °C. Kemudian nilai fraksi H₂S pada *top product* yang didapatkan berkurang menjadi 0,016788. Nilai fraksi H₂S tidak berbanding lurus dengan bertambahnya laju aliran *hot oil reboiler*. Hal ini disebabkan karena pasokan panas pada *reboiler* terlalu tinggi sehingga terjadi peningkatan pada fraksi dari komponen lain yang menyebabkan fraksi H₂S berkurang. Pada penelitian ini diharapkan fraksi H₂S pada *top product* mencapai 0,0186.

Optimisasi yang dilakukan pada penelitian ini menggunakan *genetic algorithm*. Hasil dari optimisasi yang dilakukan mendapatkan nilai terbaik untuk laju aliran *hot oil reboiler* adalah sebesar 15216823 kJ/hr dengan fraksi H₂S pada *top product* menjadi 0,018508. Pada gambar 4.3-4.6 terlihat bahwa jumlah populasi dan iterasi akan mempengaruhi waktu untuk mendapatkan nilai terbaik. Jumlah populasi dan iterasi yang banyak akan mempercepat waktu tersebut, dan sebaliknya.

Setelah didapatkan hasil optimisasi dari *genetic algorithm*, selanjutnya dilakukan pengujian dari hasil optimisasi. Berdasarkan pengujian tersebut didapatkan *set point* temperatur pada *top column* menjadi 106,35 °C. Kemudian nilai fraksi H₂S yang didapatkan dari pengujian tersebut adalah sebesar 0,018508. Nilai tersebut sudah lebih baik dari nilai awal dan sudah hampir mencapai nilai target. Sehingga dapat dikatakan bahwa *genetic algorithm* telah berhasil melakukan optimisasi.

BAB V

KESIMPULAN

Berdasarkan hasil analisis data yang telah dilakukan, didapatkan perubahan nilai laju aliran *hot oil reboiler* adalah sebesar 15216823 kJ/hr dengan nilai *set point* temperatur pada *top column* dan fraksi H₂S pada *top product* adalah masing-masing sebesar 106,35 °C dan 0,018508. Perubahan nilai laju aliran *hot oil reboiler* akan mempengaruhi kualitas pada *top product* sehingga dibutuhkan akurasi tinggi dalam penentuan nilai laju aliran tersebut.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- Ce'sar, E. A. (2013). Multi-Objective Optimization of Steam Power Plants for Sustainable. *Clean Techn Environ Policy*, 551-556.
- Cummings, A. L., Smith, G. D., & Nelsen, D. K. (2007). Advance in Amine Reclaiming-Why there's No Excuse to Operate a Dirty Amine System. *Laurance Reid Gas Conditioning Conference*.
- Haq, M. F. (2012). Optimasi Penyerapan H₂S terhadap Perubahan Suhu Ambient dalam Amine Contactor dengan Metode Pemrograman Non Linear pada Industri Pengolahan Gas Alam di Gresik. *Jurnal Teknik POMITS*, 1, 1-6.
- Kazemi, A., Malayeri, M., & Shariati, A. (2014). Feasibility Study, Simulation and Economical Evaluation of Natural Gas Sweetening Processes-Part1 1: A Case Study on A Low Capacity Plant in Iran. *Journal of Natural Gas Science and Engineering*, 20, 16-22.
- Kidnay, A. J., Parrish, W. R., & McCartney, D. G. (2011). Fundamentals of Natural Gas Processing. *CRC Press*, 218.
- Luyben, W. L. (1992). *Practical Distillation Column*. New York: VAN NOSTRAND REINHOLD.
- Marlin, T. (2000). *Process Control: Designing Processes and Control System for Dynamic Performance 2nd Edition*. New York: McGraw-Hill.
- Patil, S., & Desai, V. (2009). Optimisation of Distillation Column and Energy Reduction. *Hydrocarbon Asia*.
- Peng, D. Y., & Robinson, D. B. (1976). A New Two Constant Equation of State. *Industrial and Engineering Chemistry*, 59-94.

- Skrtic, L. (2006). *Hydrogen Sulfide, Oil and Gas, and People's Health*. Berkeley: University of California.
- Stephanopoulos, G. (1984). *Chemical Process Control*. New Jersey: P T R Prentice Hall.

LAMPIRAN

Lampiran A. Hasil Pemodelan dengan HYSYS

Tabel A.1 *Feed*

Parameter	Data Simulasi	Data 1%	Data Optimisasi
Temperatur	93.53 °C	93.53 °C	93.55 °C
Tekanan	158.0 kPa	158.6 kPa	156.3 kPa
Laju aliran massa	49494.94 kg/hr	49494.94 kg/hr	49494.94 kg/hr
Nama Komponen	Komposisi		
H2S	0.00200	0.00200	0.00200
CO2	0.03020	0.03020	0.03020
Nitrogen	0	0	0
Methane	0.00020	0.00020	0.00020
Ethane	0	0	0
Propane	0	0	0
i-Butane	0	0	0
n-Butane	0	0	0
i-Pentane	0	0	0
n-Pentane	0	0	0
n-Hexane	0	0	0
n-Heptane	0	0	0
n-Octane	0	0	0
n-Nonane	0	0	0
n-Decane	0	0	0
n-C11	0	0	0
H2O	0.86820	0.86820	0.86820
DEAmine	0.09930	0.09930	0.09930

Tabel A.2 *Reflux*

Parameter	Data Simulasi	Data 1%	Data Optimisasi
Temperatur	101.9°C	103.6 °C	102.1 °C
Tekanan	156.3 kPa	158.6 kPa	156.3 kPa
Laju aliran massa	1718 kg/hr	1582 kg/hr	241 kg/hr
Nama Komponen	Komposisi		
H2S	0	0	0.00010
CO2	0.00020	0.00020	0.00040
Nitrogen	0	0	0
Methane	0	0	0
Ethane	0	0	0
Propane	0	0	0
i-Butane	0	0	0
n-Butane	0	0	0
i-Pentane	0	0	0
n-Pentane	0	0	0
n-Hexane	0	0	0
n-Heptane	0	0	0
n-Octane	0	0	0
n-Nonane	0	0	0
n-Decane	0	0	0
n-C11	0	0	0
H2O	0.99970	0.99960	0.99920
DEAmine	0.00010	0.00010	0.00040

Tabel A.3 To Condenser

Parameter	Data Simulasi	Data 1%	Data Optimisasi
Temperatur	109.3 °C	109.9 °C	106.3 °C
Tekanan	156.3 kPa	158.6 kPa	156.3 kPa
Laju aliran massa	6617 kg/hr	6836 kg/hr	5140 kg/hr
Nama Komponen	Komposisi		
H2S	0.01240	0.01190	0.01730
CO2	0.18990	0.18030	0.26200
Nitrogen	0	0	0
Methane	0.00100	0.00100	0.00140
Ethane	0.00020	0.00020	0.00020
Propane	0.00010	0.00010	0.00020
i-Butane	0	0	0
n-Butane	0	0	0
i-Pentane	0	0	0
n-Pentane	0	0	0
n-Hexane	0	0	0
n-Heptane	0	0	0
n-Octane	0	0	0
n-Nonane	0	0	0
n-Decane	0	0	0
n-C11	0	0	0
H2O	0.79630	0.80650	0.71880
DEAmine	0	0	0

Tabel A.4 *Top Product*

Parameter	Data Simulasi	Data 1%	Data Optimisasi
Temperatur	101.9 °C	103.6 °C	102.1 °C
Tekanan	156.3 kPa	158.6 kPa	156.3 kPa
Laju aliran massa	4899 kg/hr	5258 kg/hr	4899 kg/hr
Nama Komponen	Komposisi		
H2S	0.01859	0.01679	0.01851
CO2	0.28490	0.25518	0.28026
Nitrogen	0	0	0
Methane	0.00151	0.00137	0.00151
Ethane	0.00026	0.00023	0.00025
Propane	0.00017	0.00015	0.00017
i-Butane	0.00001	0.00001	0.00001
n-Butane	0.00001	0.00001	0.00001
i-Pentane	0	0	0
n-Pentane	0	0	0
n-Hexane	0	0	0
n-Heptane	0	0	0
n-Octane	0	0	0
n-Nonane	0	0	0
n-Decane	0	0	0
n-C11	0	0	0
H2O	0.69455	0.72627	0.69928
DEAmine	0	0	0

Tabel A.5 To Reboiler

Parameter	Data Simulasi	Data 1%	Data Optimisasi
Temperatur	160.8 °C	118.4 °C	118 °C
Tekanan	171.0 kPa	171.7 kPa	169.2 kPa
Laju aliran massa	57220 kg/hr	52800 kg/hr	51390 kg/hr
Nama Komponen	Komposisi		
H2S	0	0	0
CO2	0.00020	0.00050	0.00060
Nitrogen	0	0	0
Methane	0	0	0
Ethane	0	0	0
Propane	0	0	0
i-Butane	0	0	0
n-Butane	0	0	0
i-Pentane	0	0	0
n-Pentane	0	0	0
n-Hexane	0	0	0
n-Heptane	0	0	0
n-Octane	0	0	0
n-Nonane	0	0	0
n-Decane	0	0	0
n-C11	0	0	0
H2O	0.92230	0.91280	0.90930
DEAmine	0.07760	0.08670	0.09010

Tabel A.6 *Bottom Product*

Parameter	Data Simulasi	Data 1%	Data Optimisasi
Temperatur	162.4 °C	119.5 °C	119.0 °C
Tekanan	171.0 kPa	171.7 kPa	169.2 kPa
Laju aliran massa	44600 kg/hr	44240 kg/hr	44600 kg/hr
Nama Komponen	Komposisi		
H2S	0	0	0
CO2	0.00010	0.00040	0.00050
Nitrogen	0	0	0
Methane	0	0	0
Ethane	0	0	0
Propane	0	0	0
i-Butane	0	0	0
n-Butane	0	0	0
i-Pentane	0	0	0
n-Pentane	0	0	0
n-Hexane	0	0	0
n-Heptane	0	0	0
n-Octane	0	0	0
n-Nonane	0	0	0
n-Decane	0	0	0
n-C11	0	0	0
H2O	0.88880	0.88710	0.88830
DEAmine	0.11110	0.11250	0.11120

Tabel A.7 Boil Up

Parameter	Data Simulasi	Data 1%	Data Optimisasi
Temperatur	162.4 °C	119.5 °C	119.0 °C
Tekanan	171.0 kPa	171.7 kPa	169.2 kPa
Laju aliran massa	12630 kg/hr	8560 kg/hr	6792 kg/hr
Nama Komponen	Komposisi		
H2S	0	0	0
CO2	0.00020	0.00070	0.00110
Nitrogen	0	0	0
Methane	0	0	0
Ethane	0	0	0
Propane	0	0	0
i-Butane	0	0	0
n-Butane	0	0	0
i-Pentane	0	0	0
n-Pentane	0	0	0
n-Hexane	0	0	0
n-Heptane	0	0	0
n-Octane	0	0	0
n-Nonane	0	0	0
n-Decane	0	0	0
n-C11	0	0	0
H2O	0.99930	0.99920	0.99880
DEAmine	0.00040	0.00010	0.00010

Lampiran B. Hasil Validasi Data Proses

Tabel B. 1 Validasi Data Aliran *Feed*

Parameter	Data <i>Design</i>	Data Simulasi	<i>Error</i> (%)
Temperatur	93.54 °C	93.54 °C	0
Tekanan	5.02 bar	5.02 bar	0
Laju aliran massa	49494.94 kg/hr	49494.94 kg/hr	0
Nama Komponen	Komposisi		
H2S	0.00200	0.00200	0
CO2	0.03020	0.03020	0
Nitrogen	0	0	0
Methane	0.00020	0.00020	0
Ethane	0	0	0
Propane	0	0	0
i-Butane	0	0	0
n-Butane	0	0	0
i-Pentane	0	0	0
n-Pentane	0	0	0
n-Hexane	0	0	0
n-Heptane	0	0	0
n-Octane	0	0	0
n-Nonane	0	0	0
n-Decane	0	0	0
n-C11	0	0	0
H2O	0.86820	0.86820	0
DEAmine	0.09930	0.09930	0

Tabel B. 2 Validasi Data Aliran *Reflux*

Parameter	Data <i>Design</i>	Data Simulasi	<i>Error</i> (%)
Temperatur	103.2 °C	101.9 °C	-1.26
Tekanan	0.56 bar	0.56 bar	0
Laju aliran massa	1654 kg/hr	1718 kg/hr	3.87
Nama Komponen	Komposisi		
H2S	0	0	0
CO2	0.00020	0.00020	0
Nitrogen	0	0	0
Methane	0	0	0
Ethane	0	0	0
Propane	0	0	0
i-Butane	0	0	0
n-Butane	0	0	0
i-Pentane	0	0	0
n-Pentane	0	0	0
n-Hexane	0	0	0
n-Heptane	0	0	0
n-Octane	0	0	0
n-Nonane	0	0	0
n-Decane	0	0	0
n-C11	0	0	0
H2O	0.99960	0.99970	0.01
DEAmine	0.00020	0.00010	-50

Tabel B. 3 Validasi Data Aliran *To Condenser*

Parameter	Data <i>Design</i>	Data Simulasi	<i>Error</i> (%)
Temperatur	109.2 °C	109.3 °C	0.09
Tekanan	0.56 bar	0.56 bar	0
Laju aliran massa	6610 kg/hr	6617 kg/hr	0.11
Nama Komponen	Komposisi		
H2S	0.01290	0.01240	-3.88
CO2	0.19570	0.18990	-2.96
Nitrogen	0	0	0
Methane	0.00100	0.00100	0
Ethane	0.00020	0.00020	0
Propane	0.00010	0.00010	0
i-Butane	0	0	0
n-Butane	0	0	0
i-Pentane	0	0	0
n-Pentane	0	0	0
n-Hexane	0	0	0
n-Heptane	0	0	0
n-Octane	0	0	0
n-Nonane	0	0	0
n-Decane	0	0	0
n-C11	0	0	0
H2O	0.79000	0.79630	0.80
DEAmine	0	0	0

Tabel B. 4 Validasi Data Aliran *Top Product*

Parameter	Data <i>Design</i>	Data Simulasi	Error (%)
Temperatur	103.2 °C	101.9 °C	-1.26
Tekanan	0.56 bar	0.56 bar	0
Laju aliran massa	4956 kg/hr	4899 kg/hr	-1.15
Nama Komponen	Komposisi		
H2S	0.01725	0.01859	7.77
CO2	0.26193	0.28490	8.77
Nitrogen	0	0	0
Methane	0.00140	0.00151	7.86
Ethane	0.00024	0.00026	8
Propane	0.00016	0.00017	6.25
i-Butane	0.00001	0.00001	0
n-Butane	0.00001	0.00001	0
i-Pentane	0	0	0
n-Pentane	0	0	0
n-Hexane	0	0	0
n-Heptane	0	0	0
n-Octane	0	0	0
n-Nonane	0	0	0
n-Decane	0	0	0
n-C11	0	0	0
H2O	0.71901	0.69455	-3.40
DEAmine	0	0	0

Tabel B. 5 Validasi Data Aliran *Bottom Product*

Parameter	Data <i>Design</i>	Data Simulasi	Error (%)
Temperatur	118.54 °C	119.3 °C	0.64
Tekanan	0.69 bar	0.69 bar	0
Laju aliran massa	44400 kg/hr	44340 kg/hr	-0.14
Nama Komponen	Komposisi		
H2S	0	0	0
CO2	0.00040	0.00010	-75.00
Nitrogen	0	0	0
Methane	0	0	0
Ethane	0	0	0
Propane	0	0	0
i-Butane	0	0	0
n-Butane	0	0	0
i-Pentane	0	0	0
n-Pentane	0	0	0
n-Hexane	0	0	0
n-Heptane	0	0	0
n-Octane	0	0	0
n-Nonane	0	0	0
n-Decane	0	0	0
n-C11	0	0	0
H2O	0.88740	0.88880	0.16
DEAmine	0.11210	0.11110	-0.89

Lampiran C. *Coding-an* yang digunakan pada MatLab

- Fungsi Objektif

```
function Fo=Error(x)
Qo=x(1)
```

```
Qa=18162200;
Qd=19158100;
Fa=0.017246;
Fd=0.0167882;
G=-4.2847*(10^-10);
```

```
Fo=G*(Qo-Qa)+Fa
end
```

- *Genetic Algorithm*

```
clear all
clc
```

```
convergeiter = [];
convergemax = [];
xmax = [];
ymax = [];
DFGAfit = [];
Hasilmax=[];
fitnessvector =[];
```

```
minmax = 'max';
```

```
Population = 50;
MaxGeneration = 200;
Kromosom = 15;
elitism = 0.8;
ProbCrossOver = 0.7;
ProbMutate = 0.1;
```

```
IndividuInteger = [];
eIntSc = [];
esched = [];
```

```

Datafit = [];
Data1fit = [];
DataSort = [];
ElitIndividu = [];
HMI = [];
DataFGAfit = [];
maxall = [];

Dimension = 1;
UB = [19158100];
LB = [15216703];

for i = 1:Dimension
    RangeB(i) = UB(i)-LB(i);
end

if (strcmp(minmax, 'max'))
    mm = 1;
else
    mm = -1;
end

%====Generasi Popoulasi====
Individuawal =
floor(rand(Population, (Kromosom*Dimension))+ra
nd()));
Individu = Individuawal;

%====Binary to Int====
for i = 1:Dimension
    for j = 1:Population
        IndividuInteger(j,i) =
bi2de(Individu(j, ((i*Kromosom-
Kromosom)+1):(i*Kromosom)), 'left-msb');
    end
end

Datafit = [];

for k = 1:Population

```

```

X0 = [];
cost = [];
for ii=1:Dimension
    X0(ii,1) =
floor((((IndividuInteger(k,ii))/(2^Kromosom))*
RangeB(ii))+LB(ii));
end

cost = Error(X0);
fitness = mm*cost;
Datafit = [Datafit;fitness];
[fitemax,nmax] = max(Datafit);
end

disp('GA Processing')

for Generasi = 1:MaxGeneration

    %====GA Processing====
    clc
    if (Generasi > 1)

        %====sortir====
        sort_fit =
sortrows(sort,(Kromosom*Dimension) + 1);
        Individul = sort_fit(round((1-
elitism)*Population+1):Population,:);
        Remain =
sort_fit(round(elitism*Population) +
1:Population, :);
        X = Individul;
        M = size(X,1);
        for i=1:M
            fitnessvector(i) =
X(i,(Kromosom*Dimension) + 1);
        end

        fitnessvector = fitnessvector';

        %====Setting Probability====

```

```

        for i=1:M
            Probability(i) = fitnessvector(i)
/ sum(fitnessvector);
        end

        for i=2:M
            Probability(i) = Probability(i) +
Probability(i-1);
        end

        for i = 1:M
            n=rand;
            k=1;
            for j =1:M-1
                if (n>Probability(j))
                    k = j+1;
                end
            end
            Xparents(i,:) = X(k,:);
        end

%====Crossover=====
[M,d] = size(Xparents);
Xcrossed = Xparents;
for i=1:2:M-1
    c = rand;
    if (c<=ProbCrossOver)
        p = ceil((d-1*rand));
        Xcrossed(i,:) =
[Xparents(i,1:p) Xparents(i+1,p+1:d)];
        Xcrossed(i+1,:) =
[Xparents(i+1,1:p) Xparents(i,p+1:d)];
    end
end

if (M/2~=floor(M/2))
    c = rand;
    if (c<=ProbCrossOver)
        p = ceil((d-1)*rand);
        str = ceil((M-1)*rand);

```



```

                                Xcrossed(M,:) =
[Xparents(M,1:p) Xparents(str,p+1:d)];
                                end
                                end

                                %====Mutation=====
                                [M,d] = size(Xcrossed);
                                Xnew = Xcrossed;
                                for i=1:M
                                    for j=1:d
                                        p = rand;
                                        if (p<=ProbMutate)
                                            Xnew(i,j) = 1-
Xcrossed(i,j);
                                        end
                                    end
                                end

                                %====New Population Fitness
                                Calculation=====
                                Individu =
[Xnew(:,1:(Kromosom*Dimension));Remain(:,1:(Kromosom*Dimension))];
                                end

                                ElitIndividu = [ElitIndividu; Individu];

                                for i = 1:Dimension
                                    for j = 1:Population
                                        IndividuInteger(j,i) =
bi2de(Individu(j,((i*Kromosom)-
Kromosom)+1):(i*Kromosom)), 'left-msb');
                                    end
                                end

                                Datafit = [];

                                for po = 1:Population
                                    X0 = [];
                                    cost = [];

```

```

        for ii=1:Dimension
            X0(ii,1) =
floor((((IndividuInteger(po,ii))/(2^Kromosom))
*RangeB(ii))+LB(ii));
        end

        cost = Error(X0);
        fitness = mm*cost;
        Datafit = [Datafit;fitness];
    end

    Data1fit = Datafit;
    [fitnessmax, nmax] = max(Datafit);
    DataFGAfit = [DataFGAfit;fitnessmax];
    IndividuMax = Individu(nmax,:);
    IndividuMaxLast = IndividuMax;
    Hasilmax = IndividuMax;
    sort = [Individu Datafit];
    maxall = [maxall; sort];
    for i = 1:Dimension
        HasilMaxInt(1,i) =
bi2de(Hasilmax(1,((i*Kromosom)-
Kromosom)+1):(i*Kromosom)), 'left-msb');
    end

    HMIIt = [];
    for ij=1:Dimension
        HMIIt = [HMIIt, HasilMaxInt(1,ij)];
    end

    HMI = [HMI; HMIIt];
end

plot(DataFGAfit);
hold on

[fitnessmaxf, nmaxf] = max(DataFGAfit);
for ik=1:Dimension
    X0maxfix(ik) =
floor((((HMI(nmaxf,ik)+1)/(2^Kromosom))*RangeB
(ik))+LB(ik));

```

end

```
[fitnessmaxf, nmaxf] = max(DataFGAfit)
```

```
X0maxfix
```

```
convergemax = [convergemax;fitnessmaxf];
```

```
convergeiter = [convergeiter;nmaxf];
```

```
xmax = [xmax;X0maxfix];
```

```
DFGAfit = [DFGAfit,DataFGAfit];
```

```
save
```


BIODATA PENULIS



Nama lengkap penulis adalah Hafizh Ghazidin, lahir di kota Jakarta tanggal 26 Agustus 1995. Penulis merupakan anak kedua dari lima bersaudara. Penulis telah menyelesaikan pendidikan formal, yaitu SDIT IQRO Bekasi, SMPIT Al-Kahfi Bogor, dan SMAN 67 Jakarta. Setelah lulus dari SMA, penulis diterima

di Departemen Teknik Fisika ITS. Selama kuliah, penulis telah aktif dalam organisasi dan kepanitiaan, seperti menjadi wakil kepala Departemen Eksternal di Himpunan Mahasiswa Teknik Fisika ITS periode 2015-2016 dan panitia *Engineering Physics Week* 2015 di bagian *sponsorship*. Selain itu, penulis juga mampu mempertahankan Indeks Prestasi (IP) berada di atas 3.0 hingga mampu lulus selama 8 semester dengan baik. Bagi pembaca yang memiliki kritik, saran atau ingin berdiskusi mengenai tugas akhir ini, dapat menghubungi penulis melalui email : hafizhghazidin@gmail.com